

# Group 1

Concept of Operations for  
Composable Modeling and  
Simulation

# Definition of “Component”

- (ref. Bertrand Meyers)
  1. May be used by other software elements (clients).
  2. May be used by clients without the intervention of component developers.
  3. Includes a specification of all dependencies (hardware and software platform, versions, other components).
  4. Includes a precise specification of the functionalities it offers.
  5. Is usable on the sole basis of that specification.
  6. Is easily composable with other components.
  7. Can be integrated.
- Components are NOT objects in the OO sense. (ref. Szyperski)
- Not just software (data too), not just module level. (Group)
- Open source desirable but not required. (Group)

# Acquisition of Components

- Standards may be needed
- How can acquisition be rationalized across agencies

# Motivation

- Explain why we want composability
  - Problem –
    - simulations hard to develop, hard to use
    - Hard = time, \$\$, expertise
  - Hypothesis
    - Composability will make it easier
      - Reuse
      - Assembling
      - Creating
      - Selecting
      - Recombining
      - Managing
- Distributed development ( teams, organizations)
- Tools for development

# Related Programs

## ■ CMU SEI project

- Predictable Assembly from Certifiable Components (PACC)
- Predict properties for compositions before acquiring
- Not M&S specific

## ■ BRLCAD

- Methods for building portable & modular code
- Applicable to building components with

## ■ JVB federation

- Existing components OTB, specialized “servers”

## ■ JSB

- Current emphasis fidelity
- Vision implies composability; applications, resolution
- Technical approach based on modules, no overall composability support

# Related Programs

- JBSE
- JSIMS
  - Architecture for model & module composition
- OneSAF
  - Model, module composability; various fidelity levels
  - Build different products from set of components/libraries
- JDEP
- JMASS
  - Environment, compose withy JMASS compatible
  - Collection of modules and architecture
- DARPA NICCI
  - Composability of C2 systems, real-time missions
  - Across service lines
  - Not M&S specific
- DARPA CHADS

# Use Cases for Composability

- Composite model of combat vehicle
  - Composed from sub-models of vehicle sub-systems, recompose for different apps, swap components of different fidelity
  - Training---maintenance, operator
  - Acquisition----designer
  - User—Driver(sets req) lists of components may be possible
- Integration of composable components/systems and legacy systems
  - Gateway into architecture

# Use Cases for Composability

- Composition of scenario data into specific scenarios
  - Compose training environment suitable to train specific training tasks by use trainer e.g. clearing building
  - Composed by trainer/commander
- Inter-Service joint composition of scenario environment
  - Sim/fed composers-military leaders



# Use Cases for Composability

- Composition of multiple levels of fidelity/resolution into flexible simulation environment e.g. JSB live virtual constructive
  - Operator—Commander
  - Domains — training, acquisition, experimentation, analysis

# Use Cases for Composability

- Automated decomposition
  - Dynamic decomposition of application or model
  - Redistribute execution load
  - Module level-automatic, runtime, decomposition
  - Automatic recomposition
  - Fidelity swaps

# Terms of Reference

# a. What is Composability

- (1) Ability to put together a piece of software from several components? *Yes, but too limited.*
- (2) Ability to assemble software into modules into a structure that is semantically correct wrt the original design? *Yes*
- (3) Ability to assemble software modules into a structure that is functionally correct wrt original design? *Yes*
- (4) About methods and techniques that enhance the reusability of software? *No – too broad for M&S*
- (5) A way to structure systems so they can be built from reusable components, evolved quickly, and analyzed reliably? *Yes*

# a. What is Composability

- Composability applies to data, hardware, scenarios, systems and applications.
- Other characteristics of composability
  - portability
  - distributed
  - composability is facilitated by a framework.
  - composability requires a framework?
  - comes in levels (what is being composed)
  - different people perform composition

## b. How does composability affect the modeling task?

### (1) What price must be paid

- learning curve to build composable system
- more expensive to build composable
- initial cost higher – life cycle may be lower
- general purpose more expensive
- standard may ameliorate cost?
- configuration management cost lower?
- maintenance costs lower – spread over more users
- requires developer to understand network ... things outside his/her component

## b. How does composability affect the modeling task?

- What price must be paid
  - (a) Does composition make a modeler's job easier or harder?
  - Writing composable models is harder, debugging more difficult,
  - framework may make modeler's job easier.
  - (b) Does composition facilitate automation or confound it?
  - composition facilitates automation

b. How does composability affect the modeling task?

- (2) When should component technology be used
  - (Not answered –TBD)



## c. What is the potential payback for adopting component technology?

- (1) Is the ROI sufficient to justify the change in approach?
  - ROI is not the right question. Does composability help accomplish the mission?

d. What approach were you using previously?

■ Unanswered

e. Do all the components  
provide the same  
complexity, scope, level of  
functionality?

- No, Clearly. Federate e.g. ModSAF –  
Model e.g. IR sensor

f. Skill sets the same to use components and the infrastructure?

■ No, clearly. See e. above

## g. Do they usually have pedigrees?

- pedigree is certification by authoritative source or
- pedigree is development and maintenance history
- Should components have pedigrees?

(h) Is CBSE only approach for  
“plug and play” capability

- No, not the only approach. See discussion under (j)

# (i) Is Maturity of CBSE ready to be a standard for Composability?

## ■ What is needed?

- Architectures and Architecture Guidelines
- Language for Glue Code
- Method for specifying component interfaces and contracts
- Version change control and distribution
- Ownership of Components
- Identification of persistence and data storage
- Approach dealing with inheritance; shallow vs. deep

# (i) Is Maturity of CBSE ready to be a standard for Composability?(cont.)

## ■ What is needed?

- Component contract standards for M&S Composability
- Levels of Granularity for
  - Abstraction
  - Accounting
  - Compilation
  - Delivery
  - Dispute
  - Extension
  - Fault Containment
  - Instantiation
  - Loading
  - Locality
  - Maintenance
  - System Management

## ■ See list of projects for related work



# (j) What other approaches are there?

- Composability at service level (e.g., LOS Server)
  - Sign up for service without regard to implementation
  - “Outsourcing” Applied to M&S
- Extreme programming
  - Disposable software
  - Is reuse really good
- UML, CORBA, MDA address different aspects
- Open Source, OO, Code Libraries, RMOOP

## k. Who is the customer?

- Ultimately the warfighter as the beneficiary.
- Procurement agency pays the bill.
- Enterprise – many people/agencies benefit

# I. Who would the composability program transition to?

- diffuse “general good” hard to sell to specific person.
- DMSO should retain standard.
- No commercial examples of framework architecture.

# Composability R&D Projects

- **Component frameworks for composability**
- **User requirements for composability**
- **Processes and people**
- **Leverage/exploit current work**

# Composability R&D Projects

## ■ **Component frameworks for composability**

- Study component frameworks to answer:
  - What frameworks exist now (M&S and non-M&S)?
  - What standards are needed?
  - What M&S-specific features are needed?
  - Are product lines component frameworks?
  - Is a component framework required?
  - Are component frameworks and composability frameworks the same thing?

# Composability R&D Projects

- Possible project tasks
  - Develop component framework prototype(s)
  - Seek advice from framework experts
  - Compare HLA to component framework
    - What does HLA have?
    - What is HLA missing?
    - Can HLA be extended to include the missing features?
    - Can composability be achieved with HLA?
  - Analyze relationship of tool sets and development methodologies to component frameworks

# Composability R&D Projects

## ■ User requirements for composability

- Establish users needs w.r.t composability
  - What benefits do users seek?
  - What should be flexible and what standard?
  - How much ease of use is needed?
  - How can program-specific point solutions be reconciled with common requirements?
  - How much composability is needed?
  - At what level(s) is the greatest need?  
(software, scenarios, data)
  - What is the priority for composition?  
(speed, cost, validity)

# Composability R&D Projects

- Possible project tasks
  - Broad, careful requirements analysis
  - Perform outreach, present at conferences



# Composability R&D Projects

## ■ **Processes and people**

- Develop processes to support composability

- What organizational processes are needed?
- What development processes are needed?
- How can the transition to a composability-oriented M&S environment be accomplished?
- How can legacy applications be supported?
- How can M&S developers be trained to use composability?
- What certifications will be needed?
- Is CMM or something like it applicable?

# Composability R&D Projects

## ■ **Leverage/exploit current work**

- Identify relevant current work and determine how to use it
  - What components/frameworks/architectures have been developed that would be useful?
- Possible project tasks
  - Analyze existing projects
    - JSB, JVB, ModISE, DMIT, AMC RDEC
  - Extract reusable aspects
    - Components, frameworks, methods, tools
    - M&S environment be accomplished

# Additional Thoughts

- Focus stresses module/CBSE/Component composability
  - Too narrow; data and scenarios are big issues
- It is assumed that CBSE component based simulation is the solution?
- End user interface, ease of use, must be considered for composition environment